ConTrail: Contents Delivery System Based on a Network Topology Aware Probabilistic Replication Mechanism

Yoshiaki Sakae
NEC Corporation
sakae@bc.jp.nec.com

Masumi Ichien
NEC Corporation
m-ichien@cd.jp.nec.com

Yasuo Itabashi
NEC Corporation
y-itabashi@cw.jp.nec.com

Takayuki Shizuno
NEC Corporation
t-shizuno@bc.jp.nec.com

Toshiya Okabe
NEC Corporation
t-okabe@bx.jp.nec.com

Abstract

There are some difficulties in developing a highly efficient content delivery system: where to deploy contents or replicas, real-time event handling, and selecting optimal network paths in terms of QoS and costs. We tackle these issues by combining a probabilistic contents/replicas deployment mechanism, a real-time event notification system and OpenFlow technology. The results of simulations show that our approach can reduce the average latency for contents acquisition more than existing approaches. Moreover, we describe how to achieve the QoS and save network costs by choosing suitable communication flow with OpenFlow.

1. Introduction

In addition to traditional contents created by professional content providers, CGM (Consumer Generated Media) [1] which are produced by non-professional end-users are increasing rapidly due to technological evolution such as the spread of video and audio recording devices, the spread of smart phones, a decrease in barriers to contents publicizing and so on[1]. The diversity of contents is increasing as well. To accommodate such circumstances, the importance of an efficient contents delivery platform has been increasing.

The contents delivery platforms for the CGM era have to cope with these issues:

- The difficulty of contents deployment (planning) caused by ubiquitous contents production and consumption. A content usage pattern may differ geographically because its demand may vary due to local trends. For instance, it sometimes reveals the “local production for local consumption” type usage pattern.
- It is hard to afford the space for all contents in a single storage system, so there is need for a mechanism which utilizes several storage systems cooperatively to handle huge amounts of contents.
- Quite frequent contents creation and update: It is necessary to have a high performance event notification system.
- QoS (Quality of Service), Cost of network: It is important for contents delivery platforms to consider QoS and cost of network by nature, because contents delivery service inherently tends to occupy network bandwidth for its primary purpose.

We propose a new content delivery system (ConTrail) to overcome the above issues, with the features described in 2.1.

2. ConTrail (Contents Trailing)

2.1. Overview of ConTrail

ConTrail consists of the following building blocks and is typically structured as depicted in Figure 1.

- USC is a reliable distributed file system running on a storage cluster in a DC (Data Center). We assume the size of total contents is so big that one USC cannot have enough capacity for them solely.
- NUSC manages the USCs running on DCs and organizes a tree-like overlay network according to the size of each DC. It provides VFS interface for applications such as a media server. When a media server cannot find a specific content item, NUSC forwards the content from one USC to another USC along the overlay network paths and probabilistically makes and deploys replicas of the content at USCs on the way to the destination.
- Media Server (not shown in Figure 1) is a streaming server running on NUSC. It reads contents from NUSC and sends streaming media to end-user clients.
- RENS is an event notification system. It immediately notifies only selected users of

[1] also known as UGC (User-Generated Content) or UCC (User-Created Content).

changes of the state (event) of the contents through overlay trees which will be changed in topology dynamically responding to the state of contents[2]. The RENS is used for location management of contents, notification of state of contents, and notification of NUSC information. The RENS's topology of overlay network will differ from that of NUSC.

- OpenFlow makes switches and routers programmable by providing a standardized programming interface[3]. We utilize OpenFlow to separate two different communication flows, namely the contents forwarding flow of NUSC and event messages of RENS, in order to achieve QoS and save on network costs.

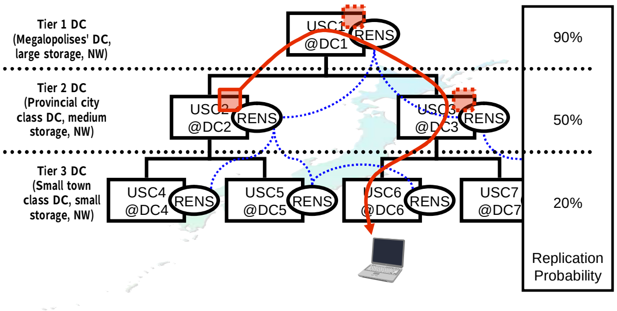

Figure 1. General structure of ConTrail

2.2. Using ConTrail

We describe the behavior of ConTrail in this section. When ConTrail is structured as depicted in Figure 1 and an end-user connected to DC6 requests the content A only stored on DC2, ConTrail will operate as follows.

1. The end-user makes a request to a media server running on DC6 for the content A.
2. The media server accesses the NUSC running on DC6 with the VFS interface for the content A.
3. The NUSC fails to find the content A in the USC running on DC6, and it inquires of RENS the location of content A. If OpenFlow configures the communication flow correctly, the query message to RENS may not heavily suffer from the contents forwarding flow by NUSC because these flows will be separated.
4. RENS resolves the location of the content A, assuming DC2 holds the content A here.
5. The NUSC running on DC6 requests the content A from the NUSC running on DC2.
6. The NUSC forwards content A from DC2 to DC6 through DC1 and DC3.
7. Each NUSC on the way to DC6 (DC1, DC3 and DC6) probabilistically makes and deploys a replica of the content A. Each NUSC will register itself as a replica node of the content A with RENS, if it decides to store the replica of the content A. Thereafter, this NUSC will be one of the contents holding nodes for the content A.
8. The media server running on DC6 sends content A to the end-user while the NUSC running on DC6 receives content A from DC3.

2.3. Merits of ConTrail

We summarize improvements and their grounds by ConTrail as follows.

- ConTrail is enabled to equip enough storage capacity for a large quantity and volume of contents, and to handle high frequency events efficiently.
  - ➢ NUSC stores contents among DCs in a distributed manner and provides a single file system image for applications by combining every USC running on DCs using RENS.
  - ➢ RENS supports ConTrail with the functionality of location management of contents and fast propagation of events.
  - ➢ OpenFlow enables selection of a network route for each communication flow, or NUSC flow and RENS flow, taking account of a tradeoff between performance and cost (e.g., prioritized network with low latency but high cost vs. best effort network).
- ConTrail can efficiently deploy contents to appropriate DCs at low-cost with NUSC.
  - ➢ NUSC places content to the DC with high probability at where the content is heavily accessed. As a result, inter-DCs transit of contents is reduced, so that it is expected that we can lower the network cost. It may be considered that this characteristic in network usage is compatible with the "local production for local consumption" type information usage pattern
  - ➢ It is not necessary to configure a replication probability at each DC but it is enough to configure replication probabilities for the tiers in a tree-like overlay network. This approach makes NUSC have notable characteristics for contents deployment: low overhead/cost, ease in adjusting the replication probability, and ease in following the changes in number of DCs.
  - ➢ NUSC also combines probabilistic replica placement approach with LRU (Least Recently Used) algorithm for outdated replica deletion. The content which once boomed but now does not will be deleted automatically with this strategy. Therefore, NUSC can easily follow the time-series

changes in contents popularity and adjust contents deployment.

- ConTrail can save network cost.
  - As mentioned above, OpenFlow can adequately assign communication flows to the network routes taking account of its performance characteristics and cost.
  - OpenFlow also can manage the communication route between DCs. For example, if OpenFlow can select a "peering route", which usually costs nothing, instead of a "transit route", we can reduce network cost dramatically.

3. Evaluation

In this paper, we only evaluate the performance and effectiveness of a probabilistic replica deployment mechanism of NUSC compared with other existing methods.

3.1. Related Approach

The contents deployment approaches in the existing system are categorized into the following four approaches[4][5]. We implemented our approach and these four approaches as simple simulations and conducted evaluations with them.

- Non-cooperative pull: In this approach, client requests are directed (using DNS redirection) to the media server running on their closest DC. If there is a cache miss, the DC pulls the content from the origin server. Most popular CDN providers (e.g., Akamai, Mirror Image) use this approach.
- Cooperative pull: The cooperative pull approach differs from the non-cooperative approach in the sense that the DCs cooperate with each other to get the requested content in case of a cache miss. Using a distributed index, the DCs find nearby copies of requested content and store them in the cache.
- Random push: The random push approach assigns contents to DCs randomly subject to the storage constraints.
- Popularity push: In this approach, each DC stores the most popular contents, to the extent the storage constraint allows. This approach assumes that the popularity order of contents is known in advance.

3.2. Evaluation Procedure

To measure the effectiveness of contents/replica placement approaches as mentioned, we chose "Content Transfer Time" which is a summation of time for steps 4 and 5 of ConTrail's operation described in 2.2.

We placed DCs in the major cities in Japan and configured the tree-like overlay network topology with network latency as shown in Figure 2. The other evaluation conditions are as follow: the storage capacity at each DC is uniform, the number of end-user accesses to each DC is uniform, the number of contents is 1000, and the contents selection follows Zipf's distribution (Figure 2).

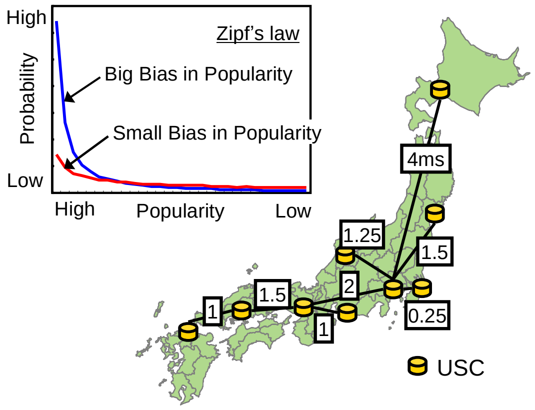

Figure 2. Evaluation conditions.

We evaluated each content placement approach as in following steps.

1. Initially place contents to DCs at random. Placement of contents and replicas is completed with push-based approaches at this point.
2. Repeat contents acquisition for enough time at each DC using each content placement approach.
3. Sum up the content transfer time.

3.3. Evaluation Results

Figure 3 and Figure 4 show the performance comparison of our proposed approach with the existing approaches described in 3.1. We evaluate our approach only with 10% of replication probability ("proposal(10)") because it shows the best performance in the preceding evaluation with 100%, 50% and 10% of probability. "nonco.pull" stands for the "Non-cooperative pull" approach. "co.pull" stands for the "Cooperative pull" approach. "co.push(rand.)" and "co.push(pop.)" stand for the "Random push" and "Popularity push" approaches respectively. The vertical axis (Average Delay) shows the average of every content transfer time that occurred in a simulation. The horizontal axis (Cache size ratio) shows the percentage of the storage capacity for replicas of contents to the total amount of contents size.

The results show that (1) co.push(rand.) shows poor performance as expected, especially when the bias in popularity is high because the random contents placement naturally expects that contents requests occur uniformly to every content item, (2) nonco.pull doesn't perform well because the contents requests have to be directed to the original contents and it involves long-distance content transfer.

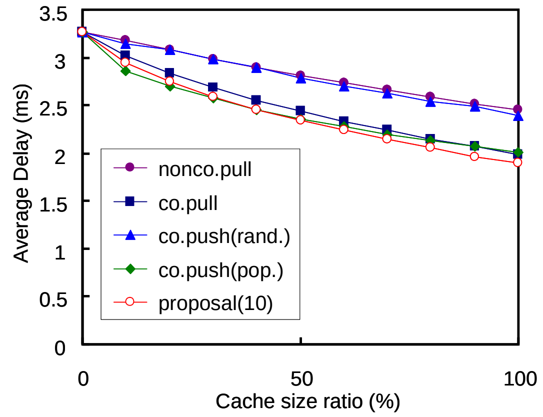

Figure 3. The performance of our approach and others' with small popularity bias.

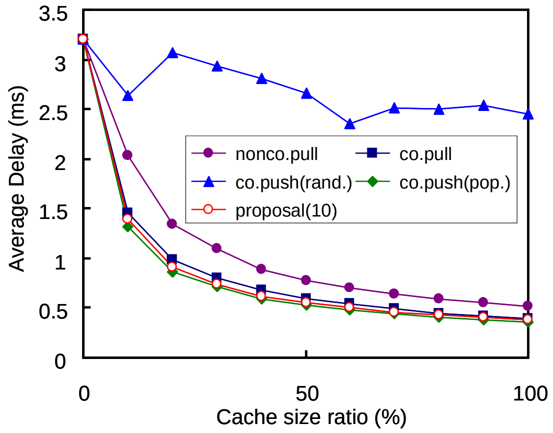

Figure 4. The performance of our approach and others' with big popularity bias.

In addition, we emphasize the following points.

- To carry out co.push(pop.) we have to grasp the popularity of all contents in advance, though it shows best performance generally.
- Our approach shows comparable performance to co.push(pop) and doesn't have to grasp the popularity of contents in advance. Therefore, our approach can follow changes in demand for contents and have high efficiency.

4. Discussion and Future work

The content and replica deployment approach we proposed as a part of NUSC can reduce the response time to receive the content and alleviate the work for planning contents placement. It also has an adequate ability to follow the changes in demand for contents. However, there is room to tune the replication probability for each tier according to run-time environment, as mentioned in 3.4. Therefore, the ease of tuning replication probability may be a key factor to applying our ConTrail easily to production use.

We invented the solution for the above issue and have applied for a patent. We would like to implement it for practical use.

5. Summary

We are developing a prototype system of ConTrail and will present an overview of the prototype as a poster session. ConTrail consists of USC, NUSC, RENS and OpenFlow. In this paper, we describe the structure of ConTrail and the performance of NUSC which is the main contribution to ConTrail.

We also show that it is possible for ConTrail to achieve efficiency and low-cost by separating communication flows which have different characteristics using OpenFlow, a key technology for new generation networks.

Acknowledgment

This research is partly supported by the National Institute of Information and Communications Technology, Japan.

References

[1] S. Wunsch-Vincent and G. Vickery, "Participative Web And User-Created Content: Web 2.0 Wikis and Social Networking, Organization for Economic, 2007.

[2] T. Shizuno, T. Kitamura, T. Okabe, and H. Tani, "Comparison of Data-Searching Algorithms for a Real-Time Information-Delivery System," Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems, 2009, pp. 430-435.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol. 38, 2008, pp. 69-74.

[4] A.K. Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks," 2007.

[5] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," Computer Communications, vol. 25, 2002, pp. 376-383.